An Approach to AP Modularization

PDES, Inc. Architecture Team
WG10 AP Modularization Workshop
Florence, Italy
October 17, 1997
Contact : David Price
dmprice@us.ibm.com
+1 (301) 803-2762

Overview of the Modular Extension Strategy

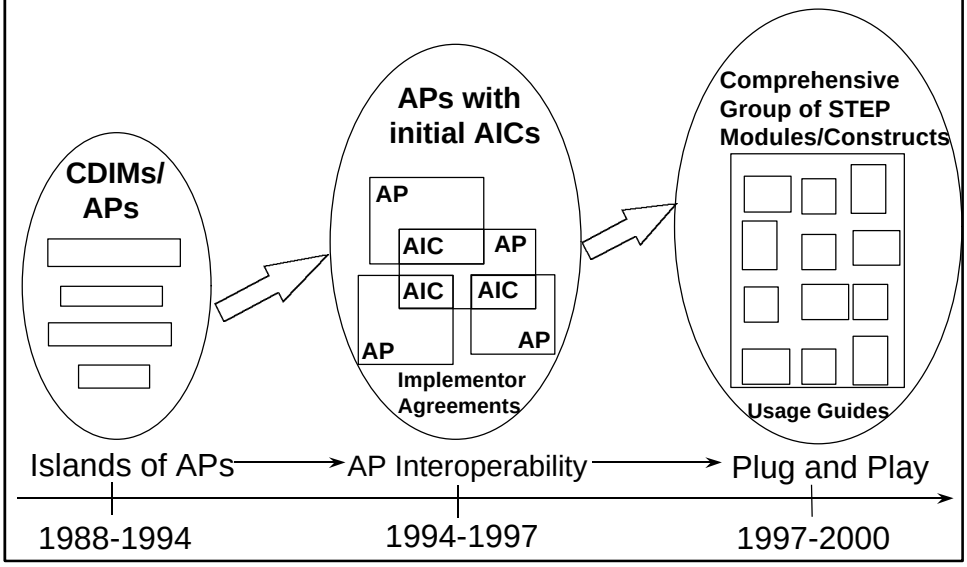
Drivers for Modularization

PDES, Inc.®

- **High cost of developing an AP / Lengthy time**
- **Companies requiring the implementation of a combination of multiples APs or AP extensions**
- **Expectation from vendors for the reuse of application software**
- **Duplication and repeated documentation of the same requirements in different APs**
- **Reuse of data generated by an implementation of one or more APs, by an implementation of one or more different APs (AP interoperability)**

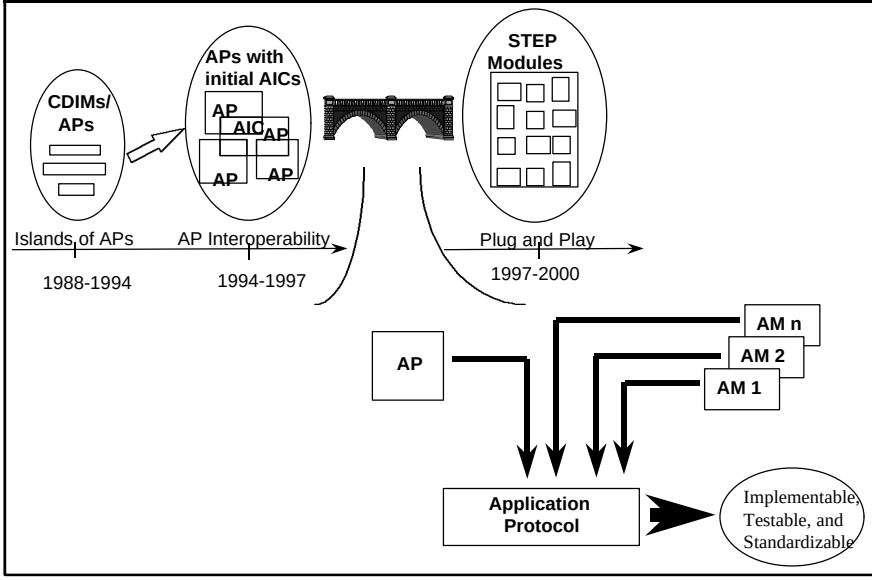
STEP Architectural Concept

PDES, Inc.®



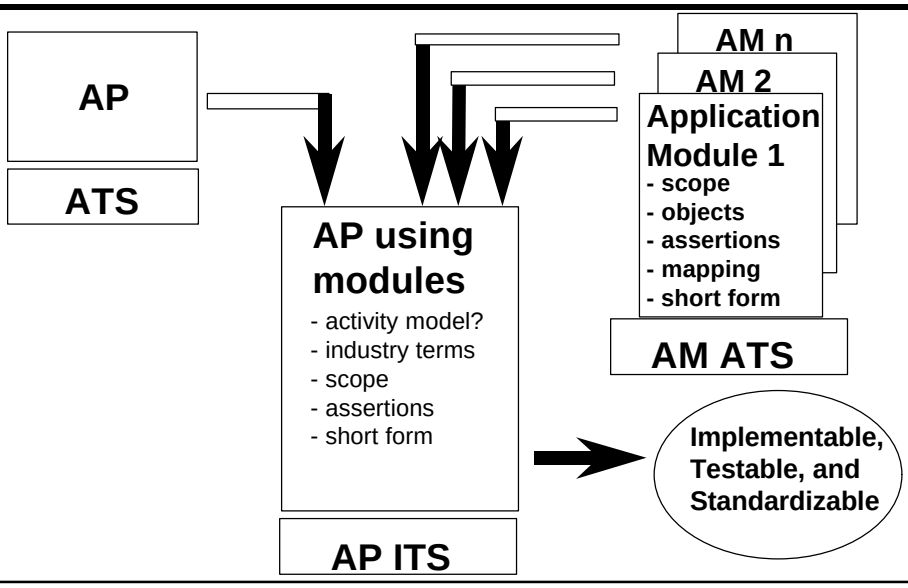
AP Module Strategy

PDES, Inc.®



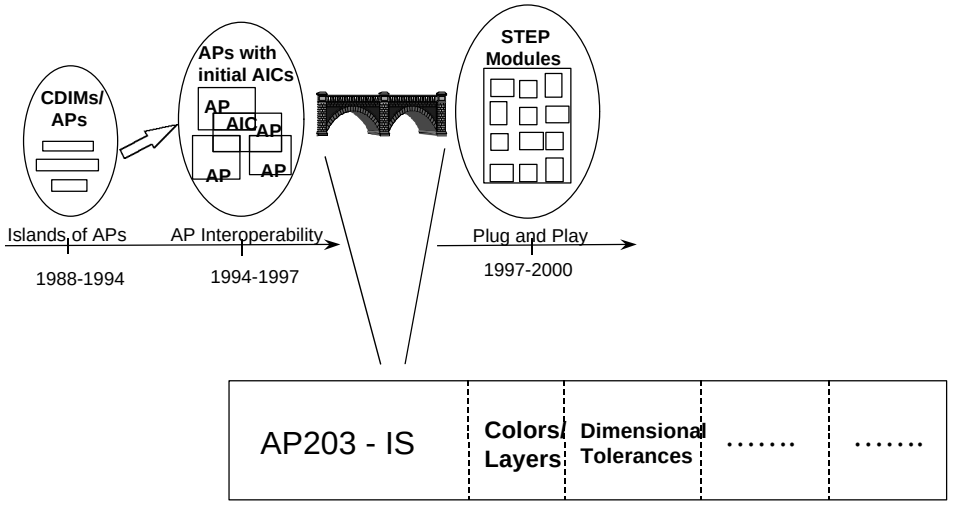
AP Modularization Approach

PDES, Inc.®



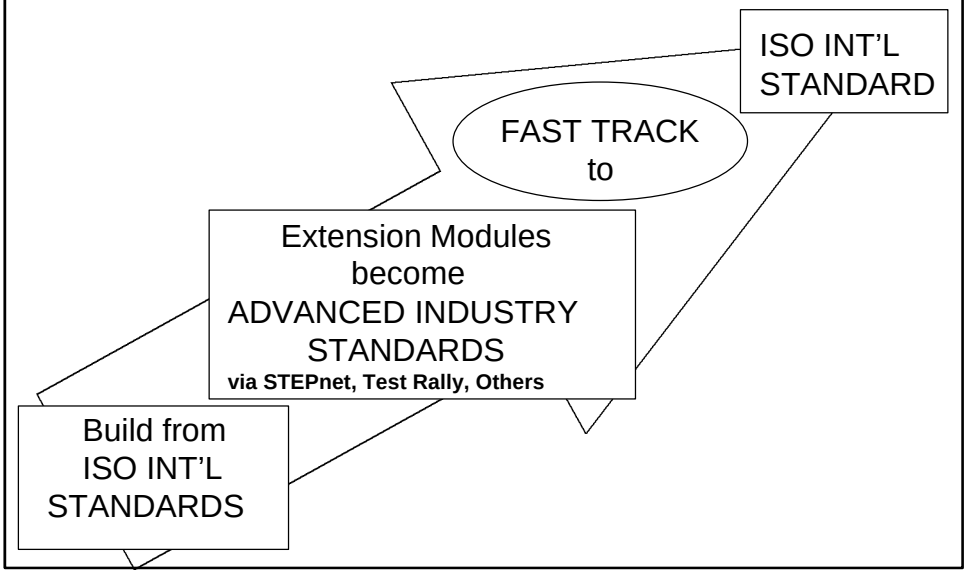
AP203 Modular Extension Strategy

PDES, Inc.®



STEP Standardization Process Improvement Concept

PDES, Inc.®



Module and AP Contents

Modules : The Next Generation AIC

- Basic objective of AIC and Application Module (AM) are quite similar
- The approach to the use and creation of an AM is different
  - Take a completely component based approach to AP development
  - Never document the same item or concept more than once
- AMs include a harmonized set of requirements, which is lacking in AICs today
- The perspective for modularization is more for implementors and users than for standards developers
- Proposing the requirement for normative EXPRESS ARMs in a module
  - This is a key aspect of the future SC4 data architecture
  - Allows use of EXPRESS-X capabilities
- Compatible with and enforces the AP Interoperability activities

Module Targets

- A module should be 100 or less pages (not a hard rule)
- A module should be a single unit of functionality
- A module is basically an AP-LET
  - It has all the components of an AP
  - This is similar to the ProSTEP proposal for AICs which would have added ARMs
  - It is different from an AIC because:
    - » an AM contains an ARM
    - » can be created as an interpretable unit even if no new rules are added
    - » does not require at least 2 APs to exist
- It may be independent or dependent
  - A module may require another module for legal implementation
    - » No one should implement colors, layers and groups by itself
  - This could be legislated by an Application Protocol
- Modules allow vendors to write code once and use it many times

AM-AIC Comparison

- 2 or more APs required
- Concepts redocumented
- Use of constructs may differ
- No ARM
- No mapping to IRs
- Not testable
- Not implementable
- Not complete
- Root node required
- All rules local to root
- Common interpretation not sufficient
- AP uses entire AIC

- A single, or no, AP required
- Concept documented only once
- A single use
- ARM documents use
- Mapping included
- Test suite based on AM
- Potentially implementable
- Complete
- No artificial root nodes
- Global rules allowed
- Common interpretation is sufficient
- AP uses entire AM

AM-New AP Guidelines Comparison

PDES, Inc.®

• May subtype multiple IR entities	• AM : YES	AP : NO
• May add cardinality global rules	• AM : YES	AP : YES
• May add subtype and constraint	• AM : YES	AP : NO
• May add subtype and derive	• AM : YES	AP : NO
• May complete mgmt resources	• AM : YES	AP : YES?
• May add value global rules	• AM : YES	AP : YES
• May add subtype and redeclare	• AM : YES	AP : NO

Summary

Modularized SC4 Standards

- AMs are the next generation of AICs - only more complete
- AMs will be small, complete and potentially implementable
- Future AP plus AM development should cost much less than AP development today
- AMs will reuse each other
- Approach should be extended to incorporate other SC4 standards
- Technical issues do remain
  - SELECT type completion is a problem
- This approach is designed to take advantage of coming EXPRESS-2 and EXPRESS-X capabilities

Backups

AM characteristics and guidelines

- 1) An AM is a functionally complete set of requirements and constructs which can be implemented to support common application requirements. This is similar to the definition of an AIC.
- 2) No conformance classes of an AM shall be allowed. Our approach removes the need for AM or AP conformance classes. An AP is small and one may be created where a conformance class was previously specified.
- 3) No activity model shall be specified in an AM.
- 4) A second AP with the same requirements is not required in order to create an AM as was required for the creation of an AIC. In fact, our approach is that APs consists of the use of a set of AMs and rules constraining the use of that set of AMs.
- 5) An AM may be a common interpretation of IR constructs. This was not sufficient to allow the creation of an AIC which required additional constraints resulting from the intepretation of application requirements (e.g. additional semantics beyond the IRs is not required).
- 6) An AM may be implementable. An AM must be complete.

AM characteristics and guidelines

- 7) An AM shall have an associated abstract test suite and implementations of it shall be testable.
- 8) An AM shall be used in its entirety by an AP or another AM. The same requirement was places on the use of an AIC.
- 9) An AM shall not provide a list of independent entities for which an AP must write a global rule as was required for an AIC.
- 10) Modules are not a process for the extension of the Integrated Resources.
- 11) The EXPRESS specified in an AM shall be compilable with references to IR schemas.
- 12) An AM documents the specific schema version and normative references it uses.
- 13) AMs shall be upward compatible.
- 14) AMs shall have unique names and shall be specified with a unique schema name.
- 15) AMs need to have as few rules in them as possible so they are as shareable as possible.

AM application reference model and mapping characteristics and guidelines

- 1) An AM ARM consists of the same elements as an AP today including application objects, application assertions and the mapping table.
- The use of EXPRESS-X for the mapping table will be investigated.
- 2) A normative EXPRESS schema shall be specified representing the application reference model. This would facilitate the use of EXPRESS-X.
- 3) All constraints defined in the mapping table within an AM shall appear in the EXPRESS schema. This includes the currently allowed AP mapping table constraints:
- - constraint of attribute values,
  - » - constraining paths between two entities,
  - » - subtype mandatory constraints,
  - » - supertype mandatory constraints,
  - » - constraining the assemble of relationships in a tree.
- 4) When mapping into an IR supertype hierarchy, make reference to the top of the supertype tree when applicable. The AP will then specify rules to specify a constraint requiring the use of a particular subtype.
- 5) EXPRESS-G ARM diagrams shall be included in the AM.

AM interpreted schema characteristics and guidelines

- 1) Each AM is defined as a single EXPRESS schema which contains inter-schema interfaces. Items may be interfaced from other AM schemas and IR schemas. This is similar to how an AIC schema is defined.
- 2) An AM is not required to specify a “root entity” as was required in an AIC.
- 3) Global rules may be specified in the AM. However, if a subtype of an IR entity is created within the AM, then constraints that apply globally may be specified as local rules in that entity (i.e. if there would naturally be a “root entity” in the AM then use it, if not do not create an artificial “root entity”). Global rules are currently allowed in an AP but not in an AIC. AM global rules may constrain the following as they may in an AP today:
- - supertype constraints,
  - » - cardinality constraints,
  - » referential integrity constraints,
  - » attribute domain constraints.

AM interpreted schema characteristics and guidelines

- 4) An AM may create subtypes of multiple IR constructs to support new capabilities. For example, the current AP practice of creating a subtype of characterized_object and a second IR entity would be applicable in an AM. Our approach would preclude this practice from being used in APs in order to allow that new concept to be shared with other APs.
- 5) An AM may specialize an entity by creating a subtype of it as is allowed in an AP today.
- 6) An AM may complete management resource constructs as is allowed in an AP today.
- 7) An AM may include examples and notes to further clarify the use of an IR entity. However, the currently allowed practice of clarifying the interpretation of an entity or attribute meaning with an AP by redefining its description would not be allowed. This “clarification” is seen as potentially limiting the reusability of the AM.
- 8) The specification of informal propositions would be allowed in an AM as in an AP today.

AM interpreted schema characteristics and guidelines

- 9) An AM may refine an IR entity in a subtype by the specification of constraints and by the definition of derived attributes as in an AP today.
- 10) An AM may constrain an attribute value population or cardinality with a global rule or by redeclaring the attribute in a subtype as in an AP today
- 11) No normative “long form” schema shall be specified for an AM.
- 12) An AM may not contain select types from which all elements have been pruned as was allowed in an AIC.
- 13) An AM may include the following EXPRESS constructs:
  - » - entity definitions that are subtypes of IR entities,
  - » - defined types,
  - » - global rules,
  - » - domain rules,
  - » - functions,
  - » - procedures,
  - » - constants.

AP characteristics and guidelines

- 1) An AP is not allowed to define new application objects.
- 2) An AP may specify information requirements using industry-specific terminology. In this case, a mapping of these requirements onto corresponding requirements in the modules is to be provided as part of the AP documentation.
- 3) A normative EXPRESS schema shall be specified representing the industry specific application reference model.
- 4) The only requirements that can be defined within an AP are any assertions that are applied to application objects defined within different AMs within the scope of the AP.
- 5) An AP may not include the specification of conformance classes.
- 6) The activity model is optional for an AP. This lowers the cost of AP development.
- 7) An AP shall have an associated integration test suite, which should reuse the module test suites, and implementations of it shall be testable.

AP schema characteristics and guidelines

- 1) An AP may not create subtypes of IR or AM entities. If this is required a module should be created and used by the AP.
- 2) APs may not refine IR or AM entities by defining derived attributes.
- 3) An AP may constrain an attribute value population or cardinality with a global rule.
- 4) It may be necessary for an AP to complete management resource constructs as is allowed in an AP today.
- 5) An AP may include examples and notes to further clarify the use of an IR entity. However, the currently allowed practice of clarifying the interpretation of an entity or attribute meaning with an AP by redefining its description would not be allowed.
- 6) The specification of informal propositions would be allowed in an AP as in an AP today.

AP schema characteristics and guidelines

PDES, Inc.®

- 7) No normative “long form” schema shall be specified for an AP.
- 8) An AP may include the following EXPRESS constructs:
  - » - defined types which are the completion of select types,
  - » - global rules,
  - » - functions,
  - » - procedures,
  - » - constants.

Module Contents

PDES, Inc.®

Module Contents...

Module Contents...

AP Document Content

AP Document Content

Process for Module Identification and Implementation

A Proposed Approach

- The modeling effort expended should be used to produce a top down integrated information model which would include both processes and data
- The high level modeling:
  - Can be achieved through the used of an adapted (by Rolls Royce Plc) Shlaer-Mellor Object Oriented Modeling technique
  - Will have integration planned in from the outset rather than done twice (once for process and once for data)
- The low level modeling:
  - Would be done based on a resource pool derived from the above
  - Would be some combination of resources, AICs, modules and APs

What is Shlaer-Mellor Modeling?

- First the subject matter is separated into “domains”
  - A domain has a mission and a distinct set of objects. It is the smallest unit of reuse
- Domains are connected by “bridges”
  - A bridge is the method through which one domain requests services from another domain
- Bridges connect “counterpart” objects
  - A counterpart is when two objects in two domains represent one another
- The above comprises a Domain Model

What is Shlaer-Mellor Modeling? (Cont.)

- For each domain:
  - The objects in the domain are defined
  - A “state model” is defined to show the explicit behavior of the object
  - An “object communication model” is defined to show the inter-object communications in a domain

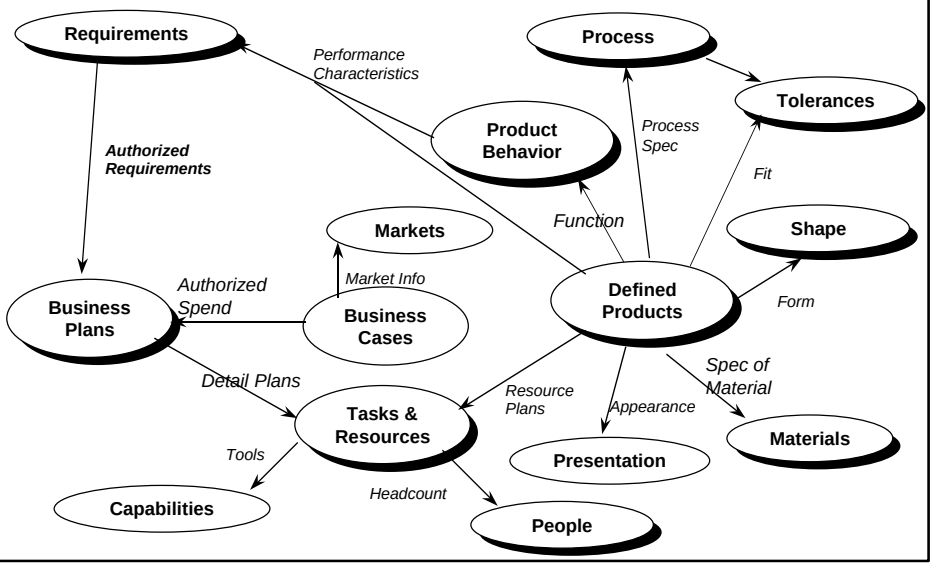
How Does This Help?

PDES, Inc.®

- **The process and data model are done as one**
- **The domain model facilitates the integration**
- **The technique is very straight forward**
- **We end up with a STEP architecture**
  - **High level objects in domains are resource models or resource entities**
  - **Low level sub-types align to modules or AICs**
  - **Implementation is factored in through the state and object communication models**
    - » **These define AMUPs or dynamic APs**

Domain Model

PDES, Inc.®



Domains

- **Business Cases**
  - **Responsible for defining opportunities with quantified demand (target market), costs, benefits, and risks**
- **Business Plans**
  - **Responsible for authorizing spend and balancing budget across the whole enterprise. The domain receives requests, evaluates them against available budget and organization initiatives, and issues orders to act or requests.**
  - **Capabilities**
  - **Responsible for non-human resources such as tools, computers, techniques, etc. which are available to the enterprise.**

Domains (Cont.)

- **Defined Products**
  - **Responsible for the definition of the enterprise product set.**
- **Materials**
  - **Responsible for defining and specifying the composition and characteristics of types of resources**
- **People**
  - **Responsible for human individuals and their associated skills which are available to the enterprise.**
- **Presentation**
  - **Responsible for defining and specifying the catalog of enterprise rendering resources and techniques.**

Domains (Cont.)

PDES, Inc.®

- **Process**
  - **Responsible for defining the possible processes with the limits of their capabilities/accuracies that can be used to produce products.**
- **Product Behavior**
  - **Responsible for defining the state models (both virtual (simulated) and actual (product test)) for products.**
- **Requirements**
  - **Responsible for formally defining and cataloging requirements for the organization**

Domains (Cont.)

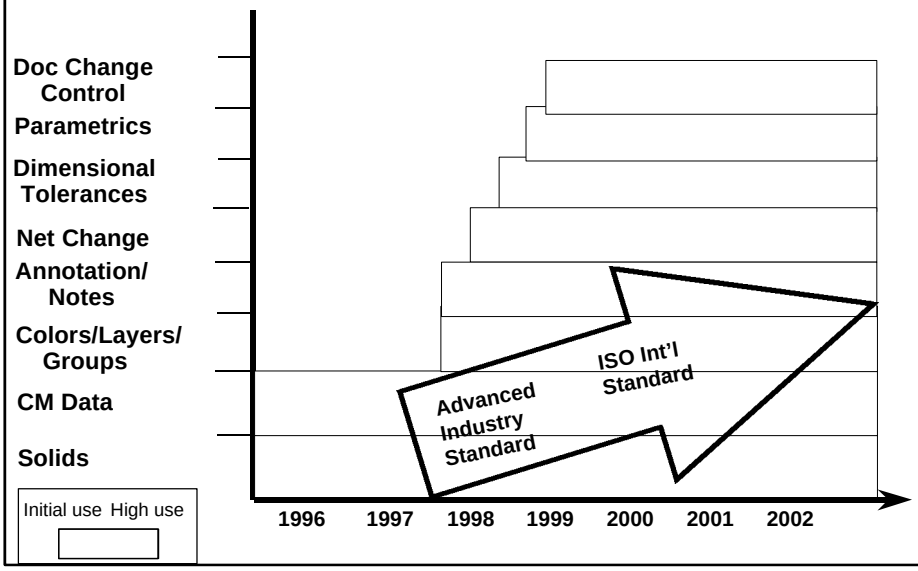
PDES, Inc.®

- **Shape**
  - **Responsible for defining the geometric, topological and spacial representation of a product or abstract notion (e.g. gas flow in a a gas turbine engine) in a given state.**
- **Tasks and Resources**
  - **Responsible for detail plans, human and tool resources, and schedules**
- **Tolerances**
  - **Responsible for defining the allowable range within which a property may deviate.**

AP 203 Extensions

Operational Use of AP 203

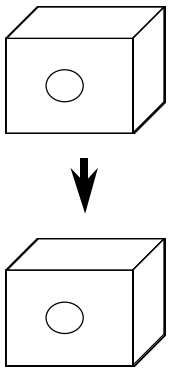
PDES, Inc.®



Colors/Layers/Groups

PDES, Inc.®

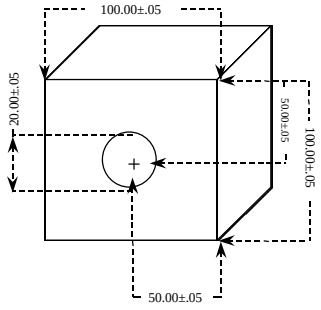
- **What is it?**
  - **Extending AP 203 to provide a geometric entity level color/layer/ group capability**
- **Why?**
  - **Extends the functionality provided by product categories in 203 today**



Dimensional Tolerances

PDES, Inc.®

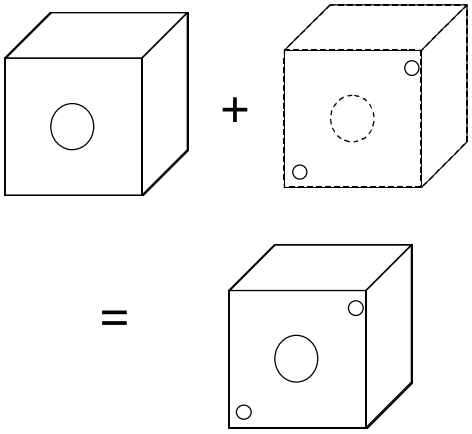
- **What is it?**
  - **Extending AP 203 to add data structures from STEP Part 47 to enable the representation of tolerances for geometric models**
- **Why?**
  - **Adds capability intended for 203 but unavailable in STEP at the time**



Net Change PDM

PDES, Inc.®

- **What is it?**
  - **Extending AP 203 to identify delta changes in a design**
- **Why?**
  - **To exchange only the information that has changed**
  - **Provides capability that PDM systems have currently**

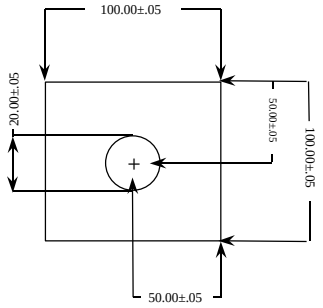


Annotation/Notes

PDES, Inc.®

- **What is it?**
  - **Extending AP 203 to provide views and textual/geometric annotation. This would add line fonts, 2D geometric annotation, text and symbols**
  - **Why?**
  - **Bring AP 203 closer to current mechanical CAD design capabilities**

Part Number: 123456-1
Nomenclature: Block with hole
Material: Plastic
CAD Model: 123456-1_Block
NHA: 22446677-1



FRONT

Parametrics

PDES, Inc.®

- **What is it?**
  - **Adding the work that results from the current STEP activities on parametric and variational geometric modeling. This would undoubtedly result in the creation of another shape representation type.**
- **Why?**
  - **Maintains design intent information**
  - **Provides support for variational and parametric modelers**

Module for Shape Appearance, Layers and Groups

Module Information

PDES, Inc.®

- **This first module is more properly 3 modules (appearance, layers and groups)**
- **If this were done:**
  - **There would be no subsections to section 4.1 (Unit of Functionality)**
  - **There would be only one mapping table versus 3**
- **The module (as it currently exists) was cut from AP 214 as that is what is implemented for the subject functionality**
- **The module required 8 days to create and one day for the first round of comment resolution**
  - **Note: this was a cut/paste/homogenize operation**
  - **Due to Word problems this could have been 6 days**
  - **A new module would require requirements definition (1-2 months)**
  - **A module built from n APs would require harmonization (@ 2 weeks)**